

CLAIMS

What is claimed is:

- 1 1. A data structure comprising:
 - 2 an object, wherein the object includes a pointer to a vtable, wherein the
 - 3 vtable includes a plurality of pointers, wherein the plurality of pointers point to a
 - 4 plurality of classes and wherein the plurality of classes include at least one unified
 - 5 type hierarchy.
- 1 2. The data structure of claim 1 wherein the unified type hierarchy includes:
 - 2 a first name from a first programming language, wherein the first name
 - 3 identifies an assigned object in the first programming language;
 - 4 a second name from a second programming language, wherein the second
 - 5 name identifies the assigned object in the second programming language; and
 - 6 a pointer to a implementation of the assigned object.
- 1 3. The data structure of claim 2 wherein if the assigned object is not identified
 - 2 in the first programming language, then the first function name is a null.
- 1 4. The data structure of claim 1 wherein the unified type hierarchy includes:
 - 2 a data structure that is recognizable by a first programming language and a
 - 3 second programming language.

1 5. The data structure of claim 1 wherein the data structure is a data structure for
2 use in two or more hierarchical programming languages.

1 6. The data structure of claim 1 wherein the data structure is a data structure for
2 use in two or more object-oriented programming languages.

1 7. The data structure of claim 6 wherein the two or more object-oriented
2 programming languages include at least two of a group consisting of:
3 Java, C# (C Sharp), C++, Smalltalk, and Eiffel.

1 8. The data structure of claim 1 further comprising:
2 a root identifying each one of a plurality of programming languages wherein
3 the data structure is recognizable in each one of the plurality of programming
4 languages.

1 9. A method of identifying equivalent data structures comprising:
2 receiving a plurality of data structures, wherein the each one of the plurality
3 of data structures are from a different one of a plurality of programming languages;
4 comparing the implementation of each one of the plurality of data structures;
5 and
6 identifying at least two of the plurality of data structures that have identical
7 implementations.

09961048-092001

1 10. The method of claim 9 wherein the plurality of programming languages
2 include at least two of a group consisting of Java, C# (C Sharp), C++, Smalltalk, and
3 Eiffel.

1 11. A computer system comprising:
2 a processor;
3 an input/output system coupled to the processor via a bus system;
4 a memory system coupled to the bus, wherein the memory system includes
5 processor executable instructions that when executed configure the processor to:
6 receive a plurality of data structures, wherein the each one of the
7 plurality of data structures are from a different one of the plurality of
8 programming languages;
9 compare the implementation of each one of the plurality of data
10 structures; and
11 identify at least two of the plurality of data structures that have identical
12 implementations.

1 12. The system of claim 11, further comprising a network adapter coupled to the
2 bus system and wherein the network adapter is coupled to a computer network.

1 13. The system of claim 11 wherein the plurality of programming languages
2 include at least two of a group consisting of Java, C# (C Sharp), C++, Smalltalk, and
3 Eiffel.

09561046-09001
T00260"340T9660

1 14. A method of unifying data structures comprising:
2 receiving a plurality of data structures, wherein the each one of the plurality
3 of data structures are from a different one of a plurality of programming languages;
4 comparing the implementation of each one of the plurality of data structures;
5 identifying at least two of the plurality of data structures that have identical
6 implementations; and
7 creating a unified data structure wherein the unified data structure includes:
8 a single implementation of the identified at least two data structures;
9 and
10 a plurality of names of the identified at least two data structures.

1 15. The method of claim 14 wherein the plurality of names of the identified at
2 least two data structures includes names that correspond to the respective one of the
3 plurality of programming languages for each of the at least two data structures.

1 16. A computer system comprising:
2 a processor;
3 an input/output system coupled to the processor via a bus system;
4 a memory system coupled to the bus, wherein the memory system includes
5 processor executable instructions that when executed configure the processor to:
6 A method of unifying equivalent data structures comprising:

7 receive a plurality of data structures, wherein the each one of the
8 plurality of data structures are from a different one of a plurality of
9 programming languages;

10 compare the implementation of each one of the plurality of data
11 structures;

12 identify at least two of the plurality of data structures that have
13 identical implementations; and

14 create a unified data structure wherein the unified data structure
15 includes:

16 a single implementation of the identified at least two data
17 structures; and

18 a plurality of names of the identified at least two data
19 structures.

1 17. The system of claim 16 wherein the plurality of programming languages
2 include at least two of a group consisting of Java, C# (C Sharp), C++, Smalltalk, and
3 Eiffel.

1 18. The system of claim 16 wherein the plurality of names of the identified at
2 least two data structures includes names that correspond to the respective one of the
3 plurality of programming languages for each of the at least two data structures.

09961048-092001